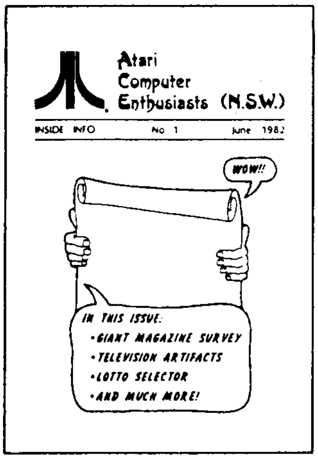
Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No. 10 December 1983

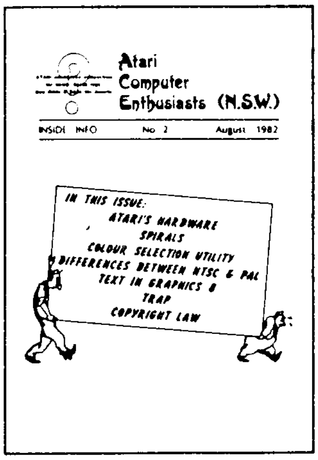


Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 1 June 1982

WOW!!

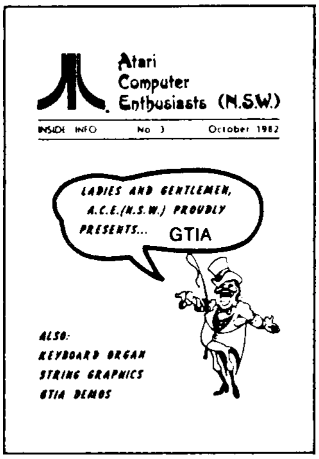
IN THIS ISSUE:
- GIANT MAGAZINE SURVEY
- TELEVISION ARTIFACTS
- LOTTO SELECTOR
- AND MUCH MORE!



Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 2 August 1982

IN THIS ISSUE:
ATARI'S HARDWARE
SPIRALS
COLOUR SELECTION UTILITY
DIFFERENCES BETWEEN NTSC & PAL
TEXT IN GRAPHICS 8
TRAP
COPYRIGHT LAW

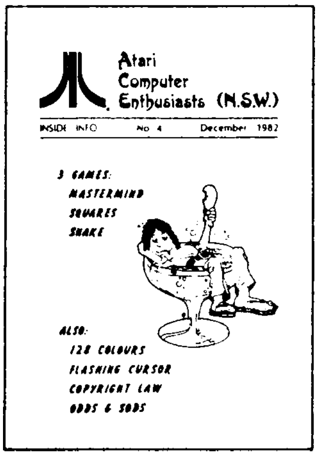


Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 3 October 1982

LADIES AND GENTLEMEN, A.C.E.(N.S.W.) PROUDLY PRESENTS... GTIA

ALSO:
KEYBOARD ORGAN
STRING GRAPHICS
GTIA DEMOS

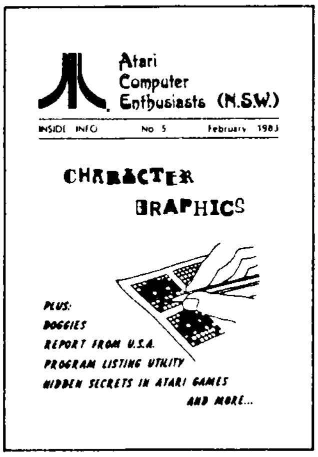


Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 4 December 1982

3 GAMES:
MASTERMIND
SQUARES
SNAKE

ALSO:
128 COLOURS
FLASHING CURSOR
COPYRIGHT LAW
ODDS & SODS

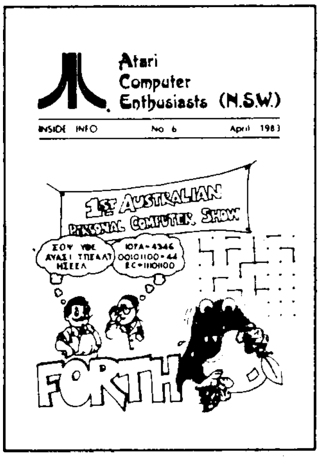


Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 5 February 1983

CHARACTER GRAPHICS

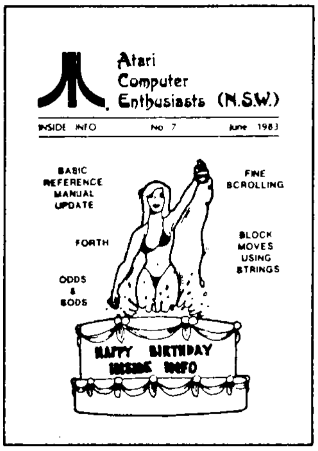
PLUS:
DOGGIES
REPORT FROM U.S.A.
PROGRAM LISTING UTILITY
HIDDEN SECRETS IN ATARI GAMES
AND MORE...



Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 6 April 1983

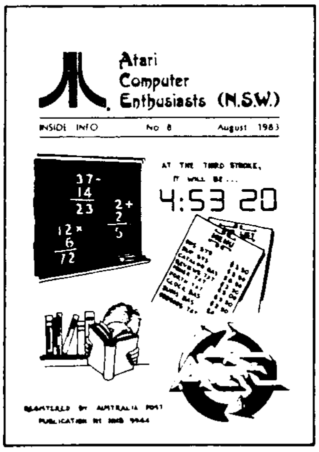
1ST AUSTRALIAN PERSONAL COMPUTER SHOW
ΣΟΥ ΥΘΕ ΑΥΑΣΙ ΤΠΕΑΛΤ ΗΙΣΕΑ
IOTA-4346 00101100-44 EC=1100100
FORTH



Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 7 June 1983

BASIC REFERENCE MANUAL UPDATE
FINE SCROLLING
FORTH
BLOCK MOVES USING STRINGS
ODDS & SODS

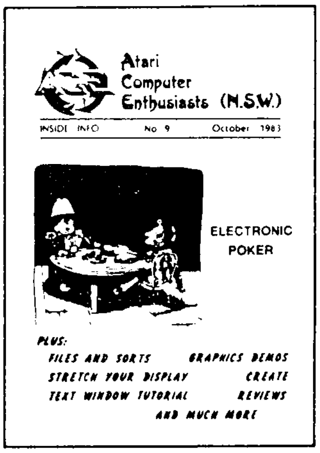


Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 8 August 1983

AT THE THIRD STROKE, IT WILL BE ...

REGISTERED BY AUSTRALIA POST PUBLICATION NO NHB 5944



Atari Computer Enthusiasts (N.S.W.)

INSIDE INFO No 9 October 1983

ELECTRONIC POKER

PLUS:
FILES AND SORTS
GRAPHICS DEMOS
STRETCH YOUR DISPLAY
CREATE
TEXT WINDOW TUTORIAL
REVIEWS
AND MUCH MORE

REGISTERED BY AUSTRALIA POST PUBLICATION No. NHB 5944

President's Annual Report

The past 12 months have seen the stabilisation and steady membership growth of A.C.E.(N.S.W.). In the following report, I will highlight significant events during the committee's term of office.

The monthly general meetings have had several venue changes, however, that has not affected the quality of presentations or software demonstrations. There were talks on wide ranging subjects which, hopefully, catered for all tastes. Our thanks go to those members who obviously spent much time and effort in preparing their subjects. I was pleased to see Futuretronics represented at recent meetings and prepared to listen to our problems.

The only organised social event was a well supported trip to see the movie TRON. In March, we participated in the 1st Australian Personal Computer Show. The area allocated to us at Centrepoint was small and out of the way, but lots of interest was shown in the club's activities. This was reflected by the number of people who joined on the spot. With the experience gained, we look forward to attending further shows.

The club's library has grown significantly and now has a wide range of books, technical manuals and magazines that can be viewed at general meetings. The Software Exchange has successfully negotiated blank media purchase at very competetive prices and this resulted in high sales of disks and cassettes. Sales of the four Software Exchange disks have been slow.

The full financial report provided by the Secretary/Treasurer shows a healthy balance. However, there are some points that must be considered. Rental of premises, with the exception of one month, has been free. Magazine printing costs were subsidised by Computer Wave for some time. Expense items already agreed to but not yet incurred include Data Perfect for the club's records and a modem for the Bulletin Board Service. Additionally, advertising income from INSIDE INFO has been poor and printing costs high, therefore careful planning needs to be done to ensure the current financial position continues.

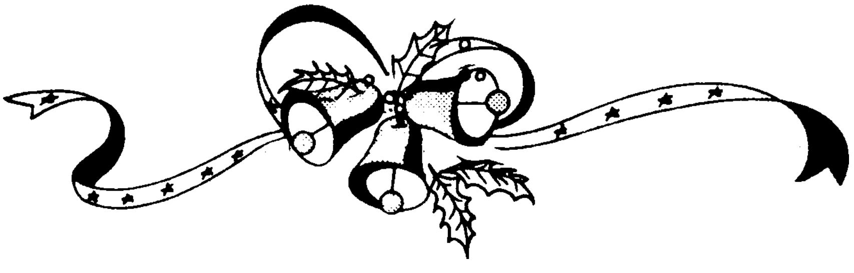
The greatest achievement has been the scheduled production of INSIDE INFO. Tremendous amounts of work are associated with collection and writing of articles, assembling the magazine and folding and posting. The quality of articles, general information and programs is exceptional. While thanking all of you who have contributed to its success, special thanks must be given to Garry Francis, who has been the driving force behind its production.

Some 12 months ago, it was decided to investigate setting up our own Bulletin Board Service. I am pleased to report that this idea is now close to reality. Futuretronics have kindly loaned us most of the hardware, as well as allowing the use of their facilities to test software. If you are unsure of what the BBS is all about, INSIDE INFO will regularly carry articles on communications concepts and the BBS.

I believe the future of A.C.E.(N.S.W.) is now assured. However, don't be complacent - participate in your club. There are many ways to help and you don't have to be an expert.

In closing, I would like to thank Bernhard Kirschner for Computer Wave's support of A.C.E.(N.S.W.) throughout the year. The committee also thanks those members who contributed in many different ways. Finally, my thanks goes to each of the committee members who gave their support and cooperation throughout the year.

- Tony Reeve



Editorial

As usual, the November meeting was our A.G.M. Following a report from the retiring President and Secretary/Treasurer, we elected a new committee for 1984. The new committee consists of:

President: Tony Reeve
Vice President: Barry Williams
Secretary/Treasurer: Anthony Dinallo
Meetings Organiser: Peter Bamford
Software Exchange (Cassette): Peter Dickeson
Software Exchange (Disk): Brian Simmons
Librarian: Jeff Maddock
BBS Operator: Roberto Romano

The more observant readers amongst you may notice that there is no Editor! I declined re-nomination, as I have some more mundane things to do - like finding money to feed myself and pay the rent. This will be my last issue of INSIDE INFO. If anybody would like to take over as Editor, please ring Tony Reeve on (02) 452-2974. In the meantime, Chris Fitzgerald has been elected (hoodwinked?) to take over on a temporary basis so that INSIDE INFO will continue uninterrupted. Please give him all the support you possibly can!

Do you realise that INSIDE INFO has been going for over 18 months? And A.C.E.(N.S.W.) has been around for almost two years? It actually seems a lot longer to me. During that period, I have published over 200 pages of material - not including ads! This includes about 60 feature programs written in BASIC, the same amount again of subroutines, program fragments and one liners also in BASIC, 5 assembly language listings, a few FORTH screens and much more! Tony has highlighted some of our other achievements from the past 12 months in his annual report.

In contrast, I thought it would be interesting to take a look at some of our non-achievements, particularly with reference to INSIDE INFO. So I took a look through all the back issues and was pleasantly surprised at how few there were. There was a number of times that I had promised an article, but it had never eventuated. On a number of occasions, this was because an article had been promised to me, but the author never fronted up with the goods. On others, I had planned writing something myself providing there was enough interest, but on each of these occasions, there was not one single response! I could only assume that the subject was of no interest, hence I would have been wasting my time to write anything. The same apathy was exhibited when we asked for help or feedback on a number of club projects. Most of these never came to fruition.

There is a good reason for pointing all this out. If you are too lazy to write articles or provide the Editor with some feedback, you will only get a second rate magazine. Keep this in mind over the next 12 months.

I have always been battled to find enough material for each issue of INSIDE INFO. In the very first issue, I said that "If every member submitted only one article or program per year, there would be sufficient material to sustain a regular newsletter". Well I've done my bit for the next 20 years. How about you? Have you submitted an article or review? Approximately 200 members haven't done their share. That's pitiful!!

Having said all that, what does the future hold? For starters, we hope to spread the Editor's massive workload amongst more people. If you can help by writing articles or reviews, typing in programs from other newsletters, folding magazines at distribution time, etc. etc. etc., then let Chris or Tony know. Just drop them a line at the usual address or see them at a meeting.

We hope to reprint more material from overseas users' groups and it would be particularly nice if we could provide a game in each issue. Any offers?

And I'll be back with two columns - just when you thought you were rid of me. There will be a regular column on Adventures and an irregular one with answers to your programming problems. The latter will be in direct response to your letters. If you have any question on programming the ATARI - any question at all - just put it in writing and send it to The Editor at our usual address. He will pass it on to me. Please be specific. I can't answer vague questions like "What does a computer do?" or "What is Player-Missile graphics?". I will probably print your question, but if you'd rather I didn't print your name, just say so. If you want a personal response, include a stamped, self addressed envelope. Also make sure to include your phone number in case I need to ring you for clarification of some point.

Anthony Dinallo has promised us some hardware articles for next issue, so watch out for that.

In closing, I'd like to give a very big thankyou to all our contributors over the last 12 months. They were (in alphabetical order) Geoff Ashworth, Jon Atack, Jamie Athas, Peter Bamford, Gavin Chapman, John Damen, Peter Dickeson, Anthony Dinallo, Peter Douglas, Peter Egan, Chris Fitzgerald, Craig Fizzell, Garry Francis, Andrew Jones, T.A. Lister, Jeff Maddock, Arthur Manrique, Greg Marthick, John Masters, John Mattes, Stan Ockers, Sean Quinlan, Gregg Ramsey, Tony Reeve, Roberto Romano, Ken Shiu, Allan Smyth, John Trigge, Brenton Vettoretti, Robert Vickery and Barry Williams. Thanks also to our friends and contacts in users' groups around the world. (Would you believe it's winter in the northern hemisphere?)

Merry Christmas and Happy New Year to you all. I'll see you next year.

-- Garry Francis

XMAS PARTY

Phillip's Foote, 101 George Street, The Rocks
6.30 P.M. Friday 9th December, 1983

For a great time, come to Sydney's famous Rocks area for the A.C.E.(N.S.W.) Christmas Party. Everybody that's anybody will be there! Bring the family and friends (but leave the toddlers at home). You should find us in the large area beside the "cookhouse". Meals average $7. Includes your choice of meat (cook it yourself), smorgasboard salad, dessert and a glass of wine or port. See you there!!

PROGRAM SELECTOR

by John Trigge

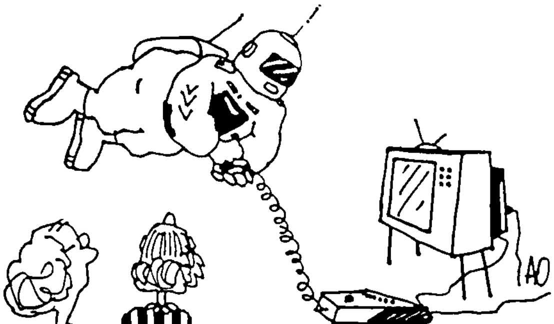
The utilities disk recently purchased from the club certainly makes writing for INSIDE INFO a lot easier. The DOC program is great.

SELECTOR will display up to 26 filenames at a time from a disk directory, each with a unique letter assigned to it. Options available are to RUN (or ENTER) a program by pressing the letter assigned to it, display more programs if there are any or to read another disk directory.

SAVEd programs will be RUN upon selection and LISTed programs will be ENTERed. I have assumed that .LST will be used as a filename extender on LISTed programs -- as recommended on page 19 of the DOS II Reference Manual. A load will be attempted for any other file selected, however an error message results if the load is unsuccessful (e.g. if trying to load DOS.SYS). Instructions and error messages are self explanatory, with a BEEP occurring whenever the message changes.

I have made this program an AUTORUN.SYS file on all my disks so that my children can easily select their program as soon as the disk boots or simply read directories until they find the program that they wish to run. This was achieved using the program from "Automate Your ATARI" by Joseph J. Wrobel in the January 1983 issue of COMPUTE! (page 146).

```
1 REM ##############################
2 REM #      PROGRAM SELECTOR       #
3 REM #       by John Trigge        #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        December 1983        #
7 REM ##############################
10 DIM FILE$(704),IN$(17),OUT$(14),T$(
1),MSG$(80),TEMP$(11):GOSUB 600:REM In
itialise
20 M=1:N=64:GOSUB 500:REM Read disk di
rectory
30 IF C=1 THEN POSITION 14,10:? " Disk
 empty ":GOTO 70
40 ? "<";CHR$(M+N);"> ";FILE$(M*11-10,
M*11)
50 IF M+N=77 THEN POKE 82,22:POSITION
22,2:REM Start second column
60 M=M+1:IF M<C AND M/27<>INT(M/27) TH
EN 40
70 POKE 82,2:GOSUB 400:REM Get message
80 GOSUB 300:REM Sound, move message a
nd get input
90 IF (A<49 OR A>50) AND (A<65 OR A>M+
63) THEN 80
100 IF A=49 AND M<C THEN GOSUB 600:N=N
-26:GOTO 40
110 IF A=49 THEN GOSUB 430:GOSUB 300:G
OTO 70:REM No more programs
120 IF A=50 THEN GOSUB 410:GOSUB 300:G
OSUB 600:GOTO 20:REM New disk
130 D=A-N:OUT$="D:"
140 TEMP$=FILE$(D*11-10,D*11-3)
150 FOR I=1 TO 8:IF TEMP$(I,I)<>" " TH
EN OUT$(LEN(OUT$)+1)=TEMP$(I,I):NEXT I
160 TEMP$=FILE$(D*11-2,D*11)
170 IF TEMP$(1,1)<>" " THEN OUT$(LEN(O
UT$)+1)=".":REM Filename extender exis
ts
180 FOR I=1 TO 3:IF TEMP$(I,I)<>" " TH
EN OUT$(LEN(OUT$)+1)=TEMP$(I,I):NEXT I
190 IF OUT$(LEN(OUT$)-2)="LST" THEN GR
APHICS 0:POKE 82,2:GOSUB 450:ENTER OUT
$
200 TRAP 210:GOSUB 450:RUN OUT$
210 GOSUB 420:GOSUB 300
220 POSITION 7,19:? "
              ":REM 30 Spaces
230 GOTO 70
300 FOR Z=1 TO 2:FOR Y=1 TO 40
310 SOUND 0,96,10,10:NEXT Y
320 SOUND 0,0,0,0:NEXT Z
330 POSITION 1,23:? MSG$(1,38);
340 T$=MSG$(1,1):MSG$=MSG$(2):MSG$(LEN
(MSG$)+1)=T$:REM Move message
350 FOR W=1 TO 40:NEXT W
360 IF PEEK(764)=255 THEN 330
370 OPEN #2,4,0,"K:":REM Open keyboard
380 GET #2,A
390 CLOSE #2:RETURN
400 MSG$="  PRESS LETTER BESIDE DESIRE
D PROGRAM  1=MORE PROGRAMS  2=NEW DISK
":RETURN
410 MSG$="  PLACE NEW DISK IN DRIVE  P
RESS ANY KEY TO CONTINUE":RETURN
420 MSG$="  PROGRAM NOT AVAILABLE FROM
 BASIC  PRESS ANY KEY TO CONTINUE":RET
URN
430 MSG$="  NO MORE FILES ON THIS DISK
  PRESS ANY KEY TO CONTINUE":RETURN
450 POSITION 7,19
460 ? "Searching for  ";OUT$
470 RETURN
500 OPEN #1,6,0,"D:*.*":REM Directory
510 C=1:FILE$=" "
520 INPUT #1,IN$:REM Read directory
530 IF IN$(5,16)="FREE SECTORS" THEN 5
70
540 IN$=IN$(3,13):REM Remove 'locked'
and sectors
550 FILE$(C*11-10,C*11)=IN$
560 C=C+1:GOTO 520
570 CLOSE #1
580 RETURN
600 POKE 82,2:GRAPHICS 0:SETCOLOR 2,1,
0:POKE 752,1
610 ? "            DISK DIRECTORY
     "
620 POKE 82,3:? :RETURN
```



...HE DOES TEND TO TAKE STAR RAIDERS A LITTLE TOO SERIOUSLY.

```
1 REM ##############################
2 REM #         AUDCTL DEMO         #
3 REM #       by Gregg Ramsey       #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        December 1983        #
7 REM ##############################
10 GRAPHICS 0:SETCOLOR 2,0,0
20 SOUND 0,1,10,8:SOUND 1,96,10,8:SOUN
D 2,162,10,8:SOUND 3,243,10,8
30 POKE 53768,0:POKE 752,1:? "USE JOYS
TICK 1 TO CHANGE AUDCTL..."
40 POSITION 9,2:? "POKE 53768,0"
50 S=STICK(0):IF S=15 THEN 50
60 P=P+(P<255 AND S=14)-(P>=1 AND S=13
)
70 POKE 53768,P:POSITION 20,2:? P;" "
80 FOR I=1 TO 20:NEXT I:GOTO 50
```

ARITHMETIC PROGRAMS

by Allan R. Smyth

All of the following programs have been used in the classroom primarily as an opportunity for the students to get hands-on experience with the computer and at the same time to DEMONSTRATE a particular lesson. They are not really intended as teaching programs, but can be of assistance in drill and remedial lessons.

With most of these programs, pressing the ESC key once provides a hint for the correct answer. In those programs which give a mark at the end, no penalty is incurred for asking for a hint. If the ESC key is pressed a second time, the answer will be given. In this case, the student is deemed to have given an incorrect answer and penalized accordingly. It has been my intention to encourage, not penalize, in doing these programs.

The following is a list of the programs with an explanation of their use. All programs use a question mark on the screen to indicate the position where a digit needs to be entered.

DIVISION

This uses the division algorithm to demonstrate long division. On RUNning the program, the first screen asks for the level. Press 1, 2, 3 or 4.

Level 1: 2 digit divisor, no remainder
Level 2: 2 digit divisor, remainder
Level 3: 3 digit divisor, no remainder
Level 4: 3 digit divisor, remainder

For the second screen, press C. The questions will now appear with a ? in the position where the digit must go. This program tests every key press and will NOT accept any incorrect digits. You may press ESC for help at any time.

MULTIPLY

The first screen asks for the level. Press 1 or 2.

Level 1: Multiply by a 2 digit number
Level 2: Multiply by a 3 digit number

For the second screen, press C. The program now runs the same as division.

IMPROPER FRACTIONS

This program converts mixed numerals to improper fractions and vice versa. Press C for both the first and second screens. Again the program uses a ? as the prompt in the required position. This program does not check each digit, but waits until the RETURN key is pressed. The ESC key provides help.

ADDITION OF FRACTIONS

This program adds two fractions. It does not check each digit, but waits until the RETURN key is pressed. An incorrect answer may be changed before the RETURN key is pressed by pressing the SPACEBAR.

EQUIVALENT FRACTIONS

This program prompts the student to enter the fourth digit in a proportion. The first screen asks for the level. Press 1 or 2.

Level 1: Equivalent fractions formed by multiplication of an integer
Level 2: Some formed by multiplication of a fraction

I could not think of a suitable hint routine for level 2, so the program does not provide help when multiplying by a non integer. Input is not checked until the RETURN key is pressed. You can change the answer by pressing the SPACEBAR.

SUMS

Screen 1 asks for the level. Press anything from 1 to 9.

Level 1: Mental arithmetic
Level 9: Very difficult

For a year 7 student requiring practice on 4 operations, probably level 3 or 4 would be most suitable. Answers are entered from right to left according to the position of the prompt. No hints are given in this program (i.e. the ESC key doesn't work). The program checks answers when the RETURN key is pressed. You may change an answer before this by pressing the SPACEBAR. If a wrong answer is given the program flashes the correct answer and loops 5 times through the same problem (typing practice?).

```
1 REM ##############################
2 REM #          DIVISION          #
3 REM #       by Allan R. Smyth    #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)    #
6 REM #        December 1983       #
7 REM ##############################
10 DIM Z$(7),ANSWER$(7):KEYS=0:KEYS1=0
20 GRAPHICS 18:OPEN #1,4,0,"K:":POSITI
ON 2,1:? #6;"         LONG":? #6:? #6;"
     DIVISION"
30 POKE 16,64:POKE 53774,64
40 POSITION 3,5:? #6;"choose level":PO
SITION 3,6:? #6;"(1,2,3 or 4)"
50 GET #1,A:IF A<49 OR A>52 THEN 50
60 LEVEL=A-48
70 ? #6;CHR$(125):? #6;"     press":?
#6
80 ? #6;"esc IF YOU NEED HELP":? #6:?
#6;"IN DOING A QUESTION"
90 POSITION 5,10:? #6;"press":? #6;"c
TO CONTINUE"
100 GET #1,A:IF A<>67 THEN 100
110 GRAPHICS 2:SETCOLOR 2,0,0
120 POKE 16,64:POKE 53774,64
130 FOR Q=1 TO 10
140 Y=INT(RND(0)*899+100)
150 IF LEVEL=1 THEN X=INT(RND(0)*7+2):
R=0
160 IF LEVEL=2 THEN X=INT(RND(0)*7+2):
R=INT(RND(0)*(X-1)+1)
170 IF LEVEL=3 THEN X=INT(RND(0)*80+11
):R=0
180 IF LEVEL=4 THEN X=INT(RND(0)*80+11
):R=INT(RND(0)*(X-1)+1)
190 Z=X*Y+R:Z$=STR$(Z):LE=LEN(Z$):Z=VA
L(Z$(1,1)):FLAG=0
200 ? #6;CHR$(125):POSITION 0,0:? #6;"
level";LEVEL
210 POSITION 9-LEN(STR$(X)),2:? #6;X;"
 ";CHR$(9);" ";Z$:POSITION 10,1:FOR I=
1 TO 9:? #6;CHR$(127);:NEXT I
220 FOR P=1 TO LE
230 ESC=0
240 IF INT(Z/X)<1 AND FLAG=0 THEN Z=10
*Z+VAL(Z$(P+1,P+1)):GOTO 740
250 FLAG=FLAG+1:ANSWER=INT(Z/X):Z=Z-AN
SWER*X
260 POSITION 0,9:? #6;"question ";Q
270 POSITION 11+P,0:? #6;CHR$(31);
280 GET #1,A:IF A=27 THEN 940
290 IF A<48 OR A>57 THEN 280
300 A=A-48:KEYS=KEYS+1
310 IF A=ANSWER THEN 340
320 ? CHR$(125);CHR$(253):KEYS1=KEYS1+
1:IF A<ANSWER THEN ? A;" IS TOO SMALL.
TRY AGAIN":GOTO 280
330 ? A;" IS TOO LARGE.TRY AGAIN":GOTO
 280
340 POSITION 11+P,0:? #6;A;:? CHR$(125
);"CORRECT.GOOD WORK"
350 ANSWER$=STR$(ANSWER*X):L=LEN(ANSWE
R$):IF P=LE AND R=0 THEN 880
360 FOR W=1 TO L
370 ESC=0
380 K=VAL(ANSWER$(L+1-W,L+1-W))
390 POSITION 12+P-W,2+FLAG
400 ? #6;CHR$(31);
410 GET #1,A:IF A=27 THEN 960
420 IF A<48 OR A>57 THEN 410
430 KEYS=KEYS+1:A=A-48
440 IF A=K THEN 460
450 KEYS1=KEYS1+1:? CHR$(125);CHR$(253
):? A;" IS INCORRECT.TRY AGAIN":GOTO 4
10
460 POSITION 12+P-W,2+FLAG:? #6;A;:? C
HR$(125);"CORRECT.PLEASE CONTINUE."
470 NEXT W
480 IF P=LE THEN 740
490 ANSWER$=STR$(Z):L=LEN(ANSWER$)
500 ESC=0
510 FOR W=1 TO L
520 K=VAL(ANSWER$(L+1-W,L+1-W))
530 POSITION 12-W+P,3+FLAG
540 ? #6;CHR$(31);
550 GET #1,A:IF A=27 THEN 980
560 IF A<48 OR A>57 THEN 550
570 A=A-48:KEYS=KEYS+1
580 IF A=K THEN 600
590 KEYS1=KEYS1+1:? CHR$(125);CHR$(253
):? "INCORRECT.TRY AGAIN":GOTO 550
600 POSITION 12-W+P,3+FLAG:? #6;A;:? C
HR$(125);"CORRECT GOOD WORK"
610 NEXT W
620 ESC=0
630 K=VAL(Z$(P+1,P+1))
640 POSITION 12+P,3+FLAG:? #6;CHR$(31)
;
650 GET #1,A:IF A=27 THEN 1000
660 IF A<48 OR A>57 THEN 650
670 A=A-48:KEYS=KEYS+1

680 IF A=K THEN 700
690 KEYS1=KEYS1+1:? CHR$(125);CHR$(253
):? "ANSWER IS INCORRECT.PLEASE TRY AG
AIN":GOTO 650
700 POSITION 12+P,3+FLAG:? #6;A;
710 ? CHR$(125);"CORRECT.GOOD WORK"
720 Z=10*Z+K
730 FLAG=FLAG+1
740 NEXT P
750 IF R=0 THEN 880
760 POSITION 11+P,0:? #6;"r";
770 ANSWER$=STR$(R):L=LEN(ANSWER$)
780 ESC=0
790 FOR W=1 TO L
800 POSITION 11+P+W,0:? #6;CHR$(31);
810 GET #1,A:IF A=27 THEN 1020
820 IF A<48 OR A>57 THEN 810
830 KEYS=KEYS+1:A=A-48
840 IF A=VAL(ANSWER$(W,W)) THEN 860
850 KEYS1=KEYS1+1:? CHR$(125);CHR$(253
):? "INCORRECT.PLEASE TRY AGAIN":GOTO
810
860 POSITION 11+P+W,0:? #6;A;
870 NEXT W
880 ? CHR$(125);"CORRECT.EXCELLENT WOR
K"
890 FOR X=1 TO 255 STEP 2:SOUND 0,RND(
0)*100+50,10,10:POKE 710,X:NEXT X:POKE
 710,0:SOUND 0,0,0,0
900 NEXT Q
910 P=INT((KEYS-KEYS1)/KEYS*100)
920 ? #6;CHR$(125):? #6;"mark ";P;"%"
930 END
940 IF ESC=0 THEN ? CHR$(125);"DIVIDE
";ANSWER*X;" BY ";X:ESC=ESC+1:GOTO 280
950 KEYS=KEYS+1:? CHR$(125):POSITION 1
1+P,0:? #6;ANSWER;:KEYS1=KEYS1+1:GOTO
350
960 IF W=1 AND ESC=0 THEN ? CHR$(125);
"TYPE THE ANSWER TO ";ANSWER;"*";X:ESC
=ESC+1:GOTO 410
970 KEYS=KEYS+1:? CHR$(125):POSITION 1
2-L+P,2+FLAG:? #6;ANSWER*X;:KEYS1=KEYS
1+1:GOTO 480
980 IF W=1 AND ESC=0 THEN ? CHR$(125);
"SUBTRACT THE ABOVE 2 NUMERALS.":ESC=E
SC+1:GOTO 550
990 KEYS=KEYS+1:? CHR$(125):POSITION 1
2-L+P,3+FLAG:? #6;Z;:KEYS1=KEYS1+1:GOT
O 620
1000 IF ESC=0 THEN ? CHR$(125);"BRING
DOWN THE NEXT DIGIT":ESC=ESC+1:GOTO 65
0
1010 KEYS=KEYS+1:? CHR$(125):POSITION
12+P,3+FLAG:? #6;K;:KEYS1=KEYS1+1:GOTO
 720
1020 IF ESC=0 AND W=1 THEN ? CHR$(125)
;"SUBTRACT THE LAST 2 NUMERALS":ESC=ES
C+1:GOTO 810
1030 KEYS=KEYS+1:POSITION 12+P,0:? #6;
R;:KEYS1=KEYS1+1:? CHR$(125):GOTO 890
```

```
1 REM ##############################
2 REM #          MULTIPLY          #
3 REM #      by Allan R. Smyth     #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)    #
6 REM #        December 1983       #
7 REM ##############################
10 DIM X$(6),Y$(3),ANSWER$(6),Y(3),X(3
),CARRY$(9):FOR I=1 TO 9:CARRY$(I)=CHR
$(16+I):NEXT I
20 KEYS=0:KEYS1=0
30 GRAPHICS 18:OPEN #1,4,0,"K:":POSITI
ON 2,1:? #6;"MULTIPLICATION"
40 POKE 16,64:POKE 53774,64
50 POSITION 3,5:? #6;"choose level":PO
SITION 5,6:? #6;"(1 or 2)"
60 GET #1,A:IF A<49 OR A>50 THEN 60
70 LEVEL=A-48
80 ? #6;CHR$(125):? #6;"     press":?
#6
90 ? #6;"esc IF YOU NEED HELP":? #6:?
#6;"IN DOING A QUESTION"
100 POSITION 5,10:? #6;"press":? #6;"c
 TO CONTINUE"
110 GET #1,A:IF A<>67 THEN 110
120 GRAPHICS 2:SETCOLOR 2,0,0
130 POKE 16,64:POKE 53774,64
140 FOR Q=1 TO 10
150 X=INT(RND(0)*986+13)
160 IF LEVEL=1 THEN Y=INT(RND(0)*86+13
):GOTO 180
170 Y=INT(RND(0)*986+13)
180 X$=STR$(X):Y$=STR$(Y):L=LEN(Y$):LE
NG=LEN(X$)
190 W=0
200 FOR Z=1 TO L
210 Y(Z)=VAL(Y$(Z,Z))
220 IF Y(Z)<>0 THEN W=W+1
230 NEXT Z
240 FOR Z=1 TO LENG
250 X(LENG+1-Z)=VAL(X$(Z,Z))
260 NEXT Z
270 ? #6;CHR$(125):POSITION 10,9:? #6;
"question";Q:POSITION 12,0:? #6;"level
";LEVEL
280 POSITION 8-LEN(X$),1:? #6;X$;"x"
290 POSITION 8-L,2:? #6;Y$
300 POSITION 2,3:? #6;"______":POSITIO
N 2,4+W:? #6;"______"
310 IF W<>1 THEN POSITION 2,6+W:? #6;"
______"
320 W=0
330 FOR Z=L TO 1 STEP -1
340 IF Y(Z)=0 THEN 420
350 ANSWER=X*Y(Z)
360 IF Z=L THEN 400
370 FOR P=1 TO (L-Z)
380 ANSWER=ANSWER*10
390 NEXT P
400 ANSWER$=STR$(ANSWER):LE=LEN(ANSWER
$)
410 GOTO 520
420 NEXT Z
430 ANSWER=X*Y:ANSWER$=STR$(ANSWER)
440 IF W=1 THEN 460
450 GOTO 1150
460 ? CHR$(125);"CORRECT.EXCELLENT WOR
K"
470 FOR X=1 TO 255 STEP 2:SOUND 0,RND(
0)*50+50,10,10:POKE 710,X:NEXT X:POKE
710,0:SOUND 0,0,0,0
480 NEXT Q
490 P=INT((KEYS-KEYS1)/KEYS*100)
500 ? #6;CHR$(125):? #6;"mark ";P;"%"
510 END
520 C=0:P=Z-L
530 FOR R=1 TO LE
540 P=P+1
550 ESC=0
560 POSITION 8-R,4+W:? #6;CHR$(31);
570 GET #1,A:IF A=27 THEN 740
580 IF A<48 OR A>57 THEN 570
590 ? CHR$(125)
600 ANSWER=A-48
610 KEYS=KEYS+1
620 IF STR$(ANSWER)=ANSWER$(LE+1-R,LE+
1-R) THEN 640
630 KEYS1=KEYS1+1:? CHR$(125);CHR$(253
):? ANSWER;" IS INCORRECT ANSWER.TRY A
GAIN":GOTO 570
640 ? "CORRECT.GOOD WORK"
650 POSITION 8-R,4+W:? #6;ANSWER;
660 IF P<1 OR P>=LENG THEN 700
670 C1=C
680 C=Y(Z)*X(P)+C
690 IF C>=10 THEN 870
700 C=0:POSITION 0,0:? #6;"            "
;
710 NEXT R
720 W=W+1:POSITION 0,0:? #6;"
 ";
730 GOTO 420
740 IF P<1 THEN 820
750 IF ESC=1 THEN KEYS=KEYS+1:KEYS1=KE
YS1+1:ANSWER=VAL(ANSWER$(LE+1-R,LE+1-R
)):? CHR$(125):GOTO 650
760 IF P>LENG THEN 980
770 ESC=ESC+1:? CHR$(125)
780 ? Y(Z);"*";X(P);
790 IF C>0 THEN ? "+";C;
800 ? "=";Y(Z)*X(P)+C
810 ? "WHAT NUMERAL DO YOU WRITE DOWN"
;:GOTO 570
820 IF ESC=1 THEN KEYS=KEYS+1:KEYS1=KE
YS1+1:ANSWER=0:GOTO 650
830 ESC=ESC+1:? CHR$(125);"REMEMBER TH
E PLACE VALUE OF ";Y(Z):? "IS ";
840 IF L-Z=1 THEN ? "TENS"
850 IF L-Z=2 THEN ? "HUNDREDS"
860 GOTO 570
870 POSITION 0,0:? #6;"           ";:C=I
NT(C/10):POSITION 7-P,0:? #6;CHR$(31);
880 ? CHR$(125);"WHAT NUMBER DO YOU CA
RRY";:ESC=0
890 GET #1,A:IF A=27 THEN 1050
900 IF A<48 OR A>57 THEN 890
910 ANSWER=A-48
920 KEYS=KEYS+1
930 IF ANSWER=C THEN 950
940 KEYS1=KEYS1+1:? CHR$(125):? ANSWER
;" IS INCORRECT.TRY AGAIN";:GOTO 890
950 POSITION 7-P,0:? #6;CARRY$(C,C);
960 ? CHR$(125)
970 GOTO 710
980 ANSWER=VAL(ANSWER$(1,2)):C=ANSWER-
Y(Z)*X(P-1)
990 ESC=ESC+1
1000 ? CHR$(125):? Y(Z);"*";X(P-1);
1010 IF C>0 THEN ? "+";C;
1020 ? "=";ANSWER
1030 ? "YOU HAVE TYPED ";VAL(ANSWER$(L
E+2-R,LE+2-R))
1040 ? "WHAT NUMBER NOW NEEDS TO BE TY
PED";:GOTO 570
1050 IF ESC=1 THEN KEYS=KEYS+1:KEYS1=K
EYS1+1:GOTO 950
1060 ESC=ESC+1:? CHR$(125):? Y(Z);"*";
X(P);
1070 IF C1>0 THEN ? "+";C1;
1080 ? "=";Y(Z)*X(P)+C1
1090 ? "YOU HAVE WRITTEN DOWN ";ANSWER
1100 ? "WHAT NUMBER DO YOU CARRY?";:GO
TO 890
1110 IF ESC=1 THEN KEYS=KEYS+1:KEYS1=K
EYS1+1:? CHR$(125):ANSWER=VAL(ANSWER$(
LE+1-R,LE+1-R)):GOTO 1250
1120 ESC=ESC+1:? CHR$(125);"ADD THE AB
OVE ";W;" NUMBERS.":?
1130 ? "CHECK FOR CARRY FROM PREVIOUS
COLUMN";
1140 GOTO 1190
1150 LE=LEN(ANSWER$)
1160 FOR R=1 TO LE
1170 ESC=0
1180 POSITION 8-R,5+W:? #6;CHR$(31);
1190 GET #1,A:IF A=27 THEN 1110
1200 IF A<48 OR A>57 THEN 1190
1210 ANSWER=A-48
1220 KEYS=KEYS+1
1230 IF STR$(ANSWER)=ANSWER$(LE+1-R,LE
+1-R) THEN 1250
1240 KEYS1=KEYS1+1:? CHR$(125);CHR$(25
3):? ANSWER;" IS INCORRECT.TRY AGAIN":
GOTO 1190
1250 ? CHR$(125);"CORRECT.GOOD WORK"
1260 POSITION 8-R,5+W:? #6;ANSWER;
1270 NEXT R
1280 GOTO 460
```

```
1 REM ###############################
2 REM #     IMPROPER FRACTIONS      #
3 REM #       by Allan R. Smyth     #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        December 1983        #
7 REM ###############################
10 CORRECT=0:GRAPHICS 18:OPEN #1,4,0,"
K:"
20 POKE 16,64:POKE 53774,64
30 POSITION 5,0:? #6;"IMPROPER"
40 POSITION 5,2:? #6;"FRACTIONS"
50 POSITION 3,10:? #6;"press c"
60 POSITION 3,11:? #6;"to continue"
70 GET #1,A:IF A<>67 THEN 70
80 GRAPHICS 18:? #6;"     press":? #6:
? #6;"esc for help":? #6:? #6;"return
when your":? #6;"answer is correct"
90 POKE 16,64:POKE 53774,64
100 POSITION 1,11:? #6;"c to continue"
110 GET #1,A:IF A<>67 THEN 110
120 GRAPHICS 2:SETCOLOR 2,0,0:POKE 16,
64:POKE 53774,64
130 FOR Q=1 TO 25:? #6;CHR$(125):ESC=0
:POSITION 9,0:? #6;"question";Q:A=32
140 CORRECT=CORRECT+1
150 DEN1=INT(RND(0)*9+2):NUM1=INT(RND(
0)*(DEN1-1)+1):WHOLE=INT(RND(0)*8+1)
160 FOR X=NUM1 TO 2 STEP -1:IF NUM1=1
THEN 190
170 IF NUM1=X*INT(NUM1/X) AND DEN1=INT
(DEN1/X)*X THEN NUM1=NUM1/X:DEN1=DEN1/
X:GOTO 160
180 NEXT X
190 NUM2=WHOLE*DEN1+NUM1
200 CHOICE=RND(0):IF CHOICE<0.5 THEN 3
40
210 GOTO 490
220 NEXT Q
230 IF CORRECT<0 THEN CORRECT=0
240 ? #6;CHR$(125):POSITION 2,2:? #6;"
MARK=__"
250 POSITION 7,1:? #6;CORRECT
260 POSITION 7,3:? #6;"25"
270 POSITION 6,5:? #6;"=";4*CORRECT;"%
"
280 POSITION 0,9
290 IF CORRECT=25 THEN ? #6;"excellent
"
300 IF CORRECT<25 AND CORRECT>20 THEN
? #6;"very good"
310 IF CORRECT<21 AND CORRECT>15 THEN
? #6;"good work"
320 IF CORRECT<16 THEN ? #6;"seek help
"
330 END
340 POSITION 4,4:? #6;NUM1
350 POSITION 3,5:? #6;WHOLE;"__=__"
360 POSITION 4,6:? #6;DEN1
370 POSITION 7,6:? #6;DEN1
```

```
380 X=0:A1=0:POSITION 7,4:? #6;"  "
390 POSITION 7+X,4:? #6;"?"
400 GET #1,A:IF A=32 THEN 380
410 IF A=27 THEN 700
420 IF A=155 AND A1<>0 THEN 460
430 IF A<48 OR A>57 THEN 400
440 POSITION 7+X,4:PUT #6,A:A1=A1*10+A
-48:X=X+1:IF X=2 THEN 460
450 GOTO 390
460 IF NUM2<>A1 THEN ? CHR$(125);CHR$(
253):? "INCORRECT.TRY AGAIN":GOTO 380
470 ? CHR$(125):? "CORRECT.GOOD WORK":
FOR A=1 TO 1000:NEXT A
480 GOTO 220
490 POSITION 4,4:? #6;NUM2
500 POSITION 4,5:? #6;"__=  __"
510 POSITION 4,6:? #6;DEN1
520 POSITION 9,6:? #6;DEN1
530 POSITION 7,5:? #6;"?"
540 GET #1,A:IF A=27 THEN 820
550 IF A<49 OR A>57 THEN 540
560 IF A-48<>WHOLE THEN ? CHR$(125):?
"INCORRECT.TRY AGAIN":? CHR$(253):GOTO
 530
570 POSITION 7,5:PUT #6,A:? CHR$(125):
? "CORRECT.GOOD WORK":ESC=0
580 POSITION 9,4:? #6;"  ":X=0:A1=0
590 POSITION 9+X,4:? #6;"?"
600 GET #1,A:IF A=32 THEN 580
610 ? CHR$(125)
620 IF A=27 THEN 750
630 IF A=155 AND A1<>0 THEN 670
640 IF A<48 OR A>57 THEN 600
650 POSITION 9+X,4:PUT #6,A:A1=A1*10+A
-48:X=X+1:IF X=2 THEN 670
660 GOTO 590
670 IF A1<>NUM1 THEN ? CHR$(125):? "IN
CORRECT .TRY AGAIN":? CHR$(253):GOTO 5
80
680 ? CHR$(125):? "CORRECT EXCELLENT W
ORK":FOR A=1 TO 1000:NEXT A
690 GOTO 220
700 IF ESC=1 THEN CORRECT=CORRECT-1:GO
TO 740
710 ESC=ESC+1
720 ? CHR$(125):? "TO FIND THE ANSWER
WE":? "MULTIPLY ";DEN1;" BY ";WHOLE;"
AND ADD ";NUM1;
730 GOTO 380
740 POSITION 7,4:? #6;"  ":POSITION 7,
4:? #6;NUM2:FOR A=1 TO 1000:NEXT A:GOT
O 220
750 IF ESC=1 THEN CORRECT=CORRECT-1:GO
TO 790
760 ESC=ESC+1
770 ? CHR$(125):? WHOLE;"*";DEN1;"=";W
HOLE*DEN1:? "HOW MANY MORE DO WE NEED
TO ADD TO ";NUM2;
780 GOTO 580
790 POSITION 9,4:? #6;"  ":POSITION 9,
4:? #6;NUM1;
800 FOR A=1 TO 1000:NEXT A
810 GOTO 220
820 IF ESC=1 THEN CORRECT=CORRECT-1:GO
TO 860
830 ESC=ESC+1
840 ? CHR$(125):? "WHAT IS ";WHOLE*DEN
1;" DIVIDED BY ";DEN1;
850 GOTO 530
860 POSITION 7,5:? #6;WHOLE;
870 ESC=0
880 GOTO 580
```

```
1 REM ###############################
2 REM #    ADDITION OF FRACTIONS    #
3 REM #      by Allan R. Smyth     #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        December 1983        #
7 REM ###############################
10 DIM C$(2):C$="+-":OPEN #1,4,0,"K:":
GRAPHICS 18
20 POKE 16,64:POKE 53774,64
30 POSITION 4,0:? #6;"ADDITION"
40 POSITION 7,1:? #6;"OF"
50 POSITION 4,2:? #6;"fractions"
60 POSITION 3,5:? #6;"press y for":? #
6;"   instructions"
70 POSITION 0,9:? #6;"press n to conti
nue"
80 GET #1,A:IF A=89 THEN GOSUB 1100:GO
TO 100
90 IF A<>78 THEN 80
100 FOR Q=1 TO 25
110 DEN1=INT(RND(0)*8+2)
120 DEN2=INT(RND(0)*8+2)
130 NUM1=INT(RND(0)*(DEN1-1)+1)
140 NUM2=INT(RND(0)*(DEN2-1)+1)
150 FOR X=NUM1 TO 2 STEP -1:IF NUM1=1
THEN 180
160 IF DEN1=X*INT(DEN1/X) AND NUM1=X*I
NT(NUM1/X) THEN DEN1=DEN1/X:NUM1=NUM1/
X:GOTO 150
170 NEXT X
180 FOR X=NUM2 TO 2 STEP -1:IF X=1 THE
N 210
190 IF DEN2=X*INT(DEN2/X) AND NUM2=X*I
NT(NUM2/X) THEN DEN2=DEN2/X:NUM2=NUM2/
X:GOTO 180
200 NEXT X
210 DEN=DEN1*DEN2:A=DEN1:IF DEN2>DEN1
THEN A=DEN2
220 FOR X=A TO 2 STEP -1
230 IF DEN=A THEN 260
240 IF DEN1=X*INT(DEN1/X) AND DEN2=X*I
NT(DEN2/X) AND DEN=X*INT(DEN/X) THEN D
EN=DEN/X
250 NEXT X
260 C=1
270 GRAPHICS 2:SETCOLOR 2,0,0
280 POKE 16,64:POKE 53774,64
290 POSITION 2,0:? #6;NUM1
300 POSITION 4,0:? #6;NUM2
310 POSITION 2,1:? #6;"_";C$(C,C);"_"
320 POSITION 2,2:? #6;DEN1
330 POSITION 4,2:? #6;DEN2
340 IF DEN1=DEN2 THEN DENA=DEN1:GOTO 4
00
350 POSITION 5,1:? #6;"=__";C$(C,C);"_
_"
360 POSITION 7,2:? #6;"?"
370 POSITION 10,2:? #6;"?"
380 GOSUB 490:? CHR$(125):IF A<>27 THE
N ? "CORRECT.GOOD WORK"
390 GOSUB 880:? CHR$(125):IF A<>27 THE
N ? "CORRECT"
400 POSITION 5,5:? #6;"=__"
410 GOSUB 640:? CHR$(125)
420 A2=DEN1*NUM2+DEN2*NUM1:A1=DEN1*DEN
2
430 IF A2=INT(A2/A1)*A1 THEN GOSUB 740
:GOTO 450
440 IF A2>A1 THEN GOSUB 775
450 ? CHR$(125):IF A<>27 THEN ? "CORRE
CT  EXCELLENT WORK";
460 FOR A=1 TO 1000:NEXT A
470 NEXT Q
480 END
490 DENA=0:X=1:ESC=0
500 IF X<3 THEN POSITION 5+X,2:? #6;"?
 ":POSITION 8+X,2:? #6;"? "
510 GET #1,A:IF A=32 THEN 490
520 IF A=27 THEN 2000
540 IF A=155 AND DENA<>0 THEN 580
550 IF A<48 OR A>57 THEN 510
560 POSITION 5+X,2:PUT #6,A:POSITION 8
+X,2:PUT #6,A:X=X+1:DENA=DENA*10+(A-48
):IF X=3 THEN 580
570 GOTO 500
580 IF DENA=DEN THEN POSITION 5+X,2:?
#6;" ":POSITION 8+X,2:? #6;" ":RETURN
590 IF DENA<>INT(DENA/DEN)*DEN THEN 63
0
600 ? CHR$(125)
610 ? "THE NUMBER ";DENA;" IS A CORREC
T":? "DENOMINATOR BUT NOT THE LOWEST "
:? "COMMON DENOMINATOR.TRY AGAIN";
620 ? CHR$(253);:GOTO 490
630 ? CHR$(125):? "DENOMINATOR INCORRE
CT":? CHR$(253):GOTO 490
640 POSITION 6,6:? #6;DENA
650 X=1:NUMA=0:ESC=0
660 IF X<4 THEN POSITION 5+X,4:? #6;"?
 "
670 GET #1,A:IF A=32 THEN 650
680 IF A=27 THEN 3000
690 IF A=155 AND NUMA<>0 THEN 720
700 IF A<48 OR A>57 THEN 670
710 POSITION 5+X,4:PUT #6,A:X=X+1:NUMA
```

```
=NUMA*10+(A-48):IF X=3 THEN 720
715 GOTO 660
720 IF NUMA=NUM1*DENA/DEN1+NUM2*DENA/D
EN2 THEN POSITION 5+X,4:? #6;" ":RETUR
N
730 ? CHR$(125):? "INCORRECT ADDITION"
:? CHR$(253):GOTO 650
740 POSITION 5,8:? #6;"=?"
750 GET #1,A:IF A=49 OR A=27 THEN POSI
TION 6,8:? #6;"1":RETURN
760 IF A>47 AND A<58 THEN ? CHR$(125):
? "INCORRECT":GOTO 740
770 GOTO 740
775 A1=0:X=1:ESC=0
780 POSITION 5,8:? #6;"=1":POSITION 7,
8:? #6;"__":POSITION 7,9:? #6;DENA
790 POSITION 6+X,7:? #6;"?  "
800 GET #1,A:IF A=32 THEN 780
810 IF A=27 THEN 4000
820 IF A=155 AND A1<>0 THEN 850
830 IF A<48 OR A>57 THEN 800
840 POSITION 6+X,7:PUT #6,A:X=X+1:A1=A
1*10+(A-48):GOTO 790
850 IF A1<>NUMA-DENA THEN ? CHR$(125):
? "INCORRECT SIMPLIFICATION":? CHR$(25
3):GOTO 775
860 POSITION 6+X,7:? #6;" "
870 RETURN
880 NUMA=0:X=1:ESC=0
890 IF X<3 THEN POSITION 5+X,0:? #6;"?
 "
900 GET #1,A:IF A=32 THEN 880
910 IF A=27 THEN 5000
920 IF A=155 AND NUMA<>0 THEN 960
930 IF A<48 OR A>57 THEN 900
940 POSITION 5+X,0:PUT #6,A:X=X+1:NUMA
=NUMA*10+(A-48):IF X=3 THEN 960
950 GOTO 890
960 IF NUMA=NUM1*DENA/DEN1 THEN 980
970 ? CHR$(125):? "NUMERATOR INCORRECT
":? CHR$(253):GOTO 880
980 POSITION 5+X,0:? #6;" "
990 NUMA=0:X=1:ESC=0:? CHR$(125)
1000 IF X<3 THEN POSITION 8+X,0:? #6;"
? "
1010 GET #1,A:IF A=32 THEN 990
1020 IF A=27 THEN 6000
1030 IF A=155 AND NUMA<>0 THEN 1070
1040 IF A<48 OR A>57 THEN 1010
1050 POSITION 8+X,0:PUT #6,A:X=X+1:NUM
A=NUMA*10+(A-48):IF X=3 THEN 1070
1060 GOTO 1000
1070 IF NUMA=NUM2*DENA/DEN2 THEN 1090
1080 ? CHR$(125):? "NUMERATOR INCORREC
T":? CHR$(253):GOTO 990
1090 POSITION 8+X,0:? #6;" ":RETURN
1100 ? #6;CHR$(125):? #6;"    press":?
 #6
1110 ? #6;"esc FOR HELP":? #6
1120 ? #6;"return WHEN YOUR":? #6;"ANS
WER IS CORRECT":? #6
1130 ? #6;"spacebar TO CHANGE":? #6;"A
NSWER "
1140 POSITION 1,11:? #6;"c TO CONTINUE
"
1150 GET #1,A:IF A<>67 THEN 1150
1160 RETURN
2000 ESC=ESC+1:ON ESC GOTO 2020,2100,2
200
2020 ? CHR$(125):? "WHAT IS THE LOWEST
 COMMON MULTIPLE":? "OF ";DEN1;" AND "
;DEN2;"?";:DENA=0:X=1:GOTO 500
2100 ? CHR$(125):? "THE COMMON DENOMIN
ATOR CAN BE FOUND BY":? "MULTIPLYING "
;DEN1;" BY ";DEN/DEN1;" OR"
2110 ? "MULTIPLYING ";DEN2;" BY ";DEN/
DEN2;:DENA=0:X=1:GOTO 500
2200 X=1:POSITION 6,2:? #6;"         ":
POSITION 5+X,2:? #6;DEN:POSITION 8+X,2
:? #6;DEN:DENA=DEN:RETURN
3000 ESC=ESC+1:IF ESC=2 THEN 3100
3010 ? CHR$(125):? "ADD ";NUM1*DENA/DE
N1;" AND ";NUM2*DENA/DEN2:NUMA=0:X=1:G
OTO 660
3100 POSITION 6,4:NUMA=NUM1*DENA/DEN1+
NUM2*DENA/DEN2:? #6;NUMA:RETURN
4000 ESC=ESC+1:IF ESC=2 THEN 4100
4010 ? CHR$(125):? NUMA;"-";DENA;"=?";
A1=0:X=1:GOTO 780
4100 POSITION 7,7:A1=NUMA-DENA:? #6;A1
:RETURN
5000 ESC=ESC+1:IF ESC=2 THEN 5100
5010 ? CHR$(125):? "WE HAVE MULTIPLIED
 THE DENOMINATOR":? "BY ";DENA/DEN1;",
WE MUST MULTIPLY THE NUMERATOR"
5020 ? "BY THE SAME NUMBER";:NUMA=0:X=
1:GOTO 890
5100 POSITION 6,0:? #6;"   ":POSITION
6,0:NUMA=NUM1*DENA/DEN1:? #6;NUMA:GOTO
 990
6000 ESC=ESC+1:IF ESC=2 THEN 6100
6010 ? CHR$(125):? "WE HAVE MULTIPLIED
 THE DENOMINATOR":? "BY ";DENA/DEN2;",
WE MUST MULTIPLY THE NUMERATOR"
6020 ? "BY THE SAME NUMBER";:NUMA=0:X=
1:GOTO 1000
6100 POSITION 9,0:? #6;"   ":POSITION
9,0:NUMA=NUM2*DENA/DEN2:? #6;NUMA:RETU
RN
```

```
1 REM ################################
2 REM #    EQUIVALENT FRACTIONS      #
3 REM #       by Allan R. Smyth      #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)      #
6 REM #         December 1983        #
7 REM ################################
10 CORRECT=0
20 GRAPHICS 18:OPEN #1,4,0,"K:":POSITI
```

```
ON 4,1:? #6;"EQUIVALENT":POSITION 5,3:
? #6;"FRACTIONS"
30 POKE 16,64:POKE 53774,64
40 POSITION 3,5:? #6;"choose level":PO
SITION 5,6:? #6;"(1 or 2)"
50 GET #1,A:IF A<49 OR A>50 THEN 50
60 LEVEL=A-49
70 ? #6;CHR$(125):? #6;"    press":? #
6
80 ? #6;"esc KEY FOR HELP":? #6:? #6;"
return WHEN YOUR":? #6;"ANSWER IS CORR
ECT"
90 POSITION 0,11:? #6;" c TO CONTINUE"
100 GET #1,A:IF A<>67 THEN 100
110 GRAPHICS 2:SETCOLOR 2,0,0
120 POKE 16,64:POKE 53774,64
130 FOR Q=1 TO 25
140 DEN1=INT(RND(0)*11+2):NUM1=INT(RND
(0)*(DEN1-1)+1)
150 CORRECT=CORRECT+1
160 FOR X=NUM1 TO 2 STEP -1:IF NUM1=1
THEN 190
170 IF DEN1=X*INT(DEN1/X) AND NUM1=X*I
NT(NUM1/X) THEN DEN1=DEN1/X:NUM1=NUM1/
X:GOTO 160
180 NEXT X
190 SCALE=INT(RND(0)*10+1):NUM2=NUM1*S
CALE:DEN2=DEN1*SCALE
200 CHOICE=INT(RND(0)*10+LEVEL*5):? #6
;CHR$(125):ESC=0
210 POSITION 0,0:? #6;"question";Q:POS
ITION 0,9:? #6;"level";LEVEL+1;
220 ON CHOICE GOTO 580,660,740,820,900
,980,1060,1140,1220
230 IF LEVEL=1 THEN 1290
240 GOTO 580
250 ? #6;CHR$(125):POSITION 4,2:? #6;"
SCORE=__":IF CORRECT<0 THEN CORRECT=0
260 POSITION 10,1:? #6;CORRECT
270 POSITION 10,3:? #6;"25"
280 POSITION 9,5:? #6;"=";4*CORRECT;"%
"
290 POSITION 1,8
300 IF CORRECT=25 THEN ? #6;"excellent
"
310 IF CORRECT<25 AND CORRECT>20 THEN
? #6;"very good"
320 IF CORRECT<21 AND CORRECT>15 THEN
? #6;"improvement needed"
330 IF CORRECT<16 THEN ? #6;"ask for h
elp"
340 END
350 X=0:A1=0:POSITION PX,PY:? #6;"   "
360 POSITION PX+X,PY:? #6;"?"
370 GET #1,A:IF A=32 THEN 350
380 IF A=27 THEN 450
390 IF A=155 AND A1<>0 THEN 430
400 IF A<48 OR A>57 THEN 370
410 POSITION PX+X,PY:PUT #6,A:X=X+1:A1
=A1*10+A-48:IF X=3 THEN 430
420 GOTO 360
430 IF A1=ANSWER THEN ? CHR$(125):? "C
ORRECT VERY GOOD":FOR A=1 TO 1000:NEXT
 A:RETURN
440 ? CHR$(125);CHR$(253):? "INCORRECT
.TRY AGAIN.":GOTO 350
450 IF ESC=1 THEN CORRECT=CORRECT-1:GO
TO 560
460 ESC=1
470 IF CHOICE=2 OR CHOICE=4 OR CHOICE=
6 OR CHOICE=7 THEN 520
480 POSITION 0,4:? #6;SCALE;"x";
490 POSITION 0,5:? #6;"___";
500 POSITION 0,6:? #6;SCALE;"x";
510 GOTO 350
520 POSITION 11,4:? #6;"x";SCALE
530 POSITION 11,5:? #6;"___";
540 POSITION 11,6:? #6;"x";SCALE;
550 GOTO 350
560 POSITION PX,PY:? #6;"   ":POSITION
 PX,PY:? #6;ANSWER:FOR A=1 TO 1000:NEX
T A
570 RETURN
580 POSITION 4,5:? #6;"___=___"
590 POSITION 4,6:? #6;DEN1
600 POSITION 8,4:? #6;NUM2
610 POSITION 8,6:? #6;DEN2
620 PX=4:PY=4:ANSWER=NUM1
630 GOSUB 350
640 NEXT Q
650 GOTO 250
660 POSITION 4,5:? #6;"___=___"
670 POSITION 8,6:? #6;DEN1
680 POSITION 4,6:? #6;DEN2
690 POSITION 4,4:? #6;NUM2
700 PX=8:PY=4:ANSWER=NUM1
710 GOSUB 350
720 NEXT Q
730 GOTO 250
740 POSITION 4,5:? #6;"___=___"
750 POSITION 4,4:? #6;NUM1
760 POSITION 4,6:? #6;DEN1
770 POSITION 8,6:? #6;DEN2
780 PX=8:PY=4:ANSWER=NUM2
790 GOSUB 350
800 NEXT Q
810 GOTO 250
820 POSITION 4,5:? #6;"___=___"
830 POSITION 4,6:? #6;DEN2
840 POSITION 8,4:? #6;NUM1
850 POSITION 8,6:? #6;DEN1
860 PX=4:PY=4:ANSWER=NUM2
870 GOSUB 350
880 NEXT Q
890 GOTO 250
900 POSITION 4,5:? #6;"___=___"
910 POSITION 4,4:? #6;NUM1
920 POSITION 4,6:? #6;DEN1
930 POSITION 8,4:? #6;NUM2
940 PX=8:PY=6:ANSWER=DEN2
```

```
950 GOSUB 350
960 NEXT Q
970 GOTO 250
980 POSITION 4,5:? #6;"___=___"
990 POSITION 4,4:? #6;NUM2
1000 POSITION 8,4:? #6;NUM1
1010 POSITION 8,6:? #6;DEN1
1020 PX=4:PY=6:ANSWER=DEN2
1030 GOSUB 350
1040 NEXT Q
1050 GOTO 250
1060 POSITION 4,5:? #6;"___=___"
1070 POSITION 4,4:? #6;NUM2
1080 POSITION 4,6:? #6;DEN2
1090 POSITION 8,4:? #6;NUM1
1100 PX=8:PY=6:ANSWER=DEN1
1110 GOSUB 350
1120 NEXT Q
1130 GOTO 250
1140 POSITION 4,5:? #6;"___=___"
1150 POSITION 4,4:? #6;NUM1
1160 POSITION 8,4:? #6;NUM2
1170 POSITION 8,6:? #6;DEN2
1180 PX=4:PY=6:ANSWER=DEN1
1190 GOSUB 350
1200 NEXT Q
1210 GOTO 250
1220 POSITION 7,5:? #6;"=___"
1230 POSITION 4,5:? #6;DEN1
1240 POSITION 8,6:? #6;SCALE
1250 PX=8:PY=4:ANSWER=DEN2
1260 GOSUB 350
1270 NEXT Q
1280 GOTO 250
1290 SCALE2=INT(RND(0)*10+1):IF SCALE2
=SCALE THEN 1290
1300 ESC=1
1310 NUM1=NUM1*SCALE2:DEN1=DEN1*SCALE2
:CHOICE=INT(RND(0)*8+1)
1320 ON CHOICE GOTO 580,660,740,820,90
0,980,1060,1140
1330 GOTO 1140
```

```
1 REM ###############################
2 REM #            SUMS             #
3 REM #      by Allan R. Smyth      #
4 REM # Published by Atari Computer #
5 REM #    Enthusiasts (N.S.W.)     #
6 REM #        December 1983        #
7 REM ###############################
10 DIM A$(4):A$="+-*/"
20 GRAPHICS 2:OPEN #1,4,0,"K:"
30 POKE 16,64:POKE 53774,64
40 SETCOLOR 2,0,0
50 ? #6;"LEVEL OF DIFFICULTY":POSITION
 6,6:? #6;"(1-9)?"
60 GET #1,D:IF D<49 OR D>57 THEN 60
70 CORRECT=0:ANSWERS=0:D=D-48
80 ? #6;CHR$(125):POSITION 13,9:? #6;"
level";D:POSITION 9,0:? #6;"question";
ANSWERS+1
90 SUM=INT(RND(0)*4+1):ON SUM GOSUB 17
0,210,260,290
100 GOSUB 400
110 POSITION 5,7:IF X=X3 THEN ? #6;"CO
RRECT":CORRECT=CORRECT+1
120 IF X<>X3 THEN ? #6;"WRONG":? CHR$(
253):GOSUB 370
130 ANSWERS=ANSWERS+1:FOR Z=1 TO 100:N
EXT Z
140 IF ANSWERS=25 THEN ? #6;CHR$(125):
? #6;CORRECT;" CORRECT":? #6:? #6;ANSW
ERS-CORRECT;"WRONG":GOSUB 160:END
150 GOTO 80
160 FOR X=1 TO 8:FOR Z=60 TO 120:SOUND
 1,Z,10,8:NEXT Z:NEXT X:RETURN
170 X1=INT(RND(0)*(10^(D/2)+10))
180 X2=INT(RND(0)*(10^(D/2)+10))
190 GOSUB 320:X3=X2+X1
200 RETURN
210 X1=INT(RND(0)*(10^(D/2)+10))
220 X2=INT(RND(0)*(10^(D/2)+10))
230 IF X1<X2 THEN X3=X1:X1=X2:X2=X3
240 GOSUB 320:X3=X1-X2
250 RETURN
260 X1=INT(RND(0)*(10^(D/3)+8)):X2=INT
(RND(0)*(6+D/1.5))
270 GOSUB 320:X3=X1*X2
280 RETURN
290 X3=INT(RND(0)*(10+D/4)):X2=INT(RND
(0)*(5+D/1.5)+1):X1=X2*X3
300 GOSUB 320
310 RETURN
320 POSITION 6-LEN(STR$(X1)),0:? #6;X1
;A$(SUM,SUM)
330 POSITION 6-LEN(STR$(X2)),1:? #6;X2
340 POSITION 1,2:? #6;"_____"
350 POSITION 1,4:? #6;"_____"
360 RETURN
370 ? #6;"answer was ";X3:FOR Z=1 TO 5
00:NEXT Z
390 RETURN
400 Z=1:Y=1:X=0
410 ? CHR$(125);"IF YOU MAKE A MISTAKE
 PRESS SPACEBAR":? "PRESS RETURN WHEN
CORRECT";
420 POSITION 6-Z,3:? #6;"?":POSITION 6
-Z,3
430 GET #1,A:IF A=155 AND Z=1 THEN 430
440 IF A=155 THEN 490
450 IF A=32 THEN ? CHR$(125):POSITION
0,3:? #6;"      ":GOTO 400
460 IF A<48 OR A>57 THEN 430
470 PUT #6,A:X=X+VAL(CHR$(A))*Y:Z=Z+1:
Y=Y*10:IF Z=6 THEN 490
480 GOTO 420
490 RETURN
```

CASSETTE RECORDER PROBLEMS

by R. S. Vickery

Like most Atarians, I have had a lot of problems with the ATARI 410 cassette recorder. I didn't know what to do about them until Carl Evans published a series of articles on the 410 in Antic magazine. Since following his instructions (more or less), the 410 has worked perfectly.

The first step was to clean the heads. These are quite easy to get at when the PLAY key is pressed down. Careful rubbing with a swab of cotton wool wrapped around forceps and moistened with methylated spirits removed all the brown gunk.

The azimuth of the play/record head was adjusted next. There is a little hole in the case just above the azimuth screw. In the newer models of the 410, this hole is covered by the name plate. I played a commercially recorded music cassette with a POKE 54018,52 command and turned the azimuth screw backwards and forwards until the treble tones were at their loudest. The original setting was quite a long way out. The screw driver for this job must not be magnetic, that is, you shouldn't be able to pick up pins or paper clips with it.

Some of the errors in loading came when the recorder was jolted, so I opened up the plug on the cable. The rivet in the plug was drilled out and replaced by a small nut and bolt. One of the wires had come out of its pin. I crimped it back in and, for good measure, soldered all the wires to their pins.

After all this, the recorder was still unreliable, so I opened up the case and started poking about in the electronics. The circuit was pretty much as Carl Evans described it, except that all the components were of excellent quality and everything seemed to be working according to specifications.

The only problem was with the width of the pulses on the data output line. They are supposed to be 1.67 microseconds long to give 600 bits per second. In fact, the high voltage pulses (which the ATARI reads as 1) were only 1.4 microseconds and the low voltage pulses (read as 0) were 1.8 microseconds. The fault was in the multiple feedback band-pass filter for the high frequency tone. This filter should have a centre frequency at about 5.3 kHz, but the actual value was 5.6 kHz. A few calculations showed that the design was wrong. I replaced resistor R114, which was originally 240 kOhm, with a 270 kOhm resistor. This resistor is in the feedback loop of the high frequency filter. The centre frequency was brought down to 5.3 kHz and all problems disappeared.

Continuous tones on the Serial Data Out line (pin 5) are useful for testing the system. The high frequency tone can be generated by the following series of POKEs.

```
POKE 53768,80:REM AUDCTL
POKE 53760,161:REM Set N low byte on voice 1
POKE 53762,0:REM Set N high byte on voice 2
POKE 53761,175:REM AUDC1 pure tone
POKE 53763,175:REM AUDC2 pure tone
POKE 53775,139:REM SKCTL 2-tone output, keyboard enabled
```

Change to the low frequency tone with POKE 53760,217.

I still have a tape of alternating 1's and 0's which I used for testing the 410 and would be happy to lend it to anybody interested in fixing their own machine.

* * *

WANTED

Any issues of "Computer and Video Games" magazine prior to July 1982. Also March, April, June, July and August 1983.

Adventure International's Adventure #12: Golden Voyage

– Contact Garry Francis (02) 789-1397 (A.H.)

SPRITE MAKER

By John Mattes

After an absence of a couple of months, I am pleased to return to pages of ACE with an article on player-missile graphics in FORTH.

The Team Atari disk has a demo on SCR # 3A which allows you to drive a player around the screen with joystick. For those who haven't tried it, boot your working disk, (which must already have SCR # 34 and 35 loaded), insert your source disk and type:

```
3A   LOAD   SPLAY   RUNIT
```

To create your own players, use the word "SPRITE", the code for which is listed in SCR # 18. SPRITE is word which can create as many different players as you want. For example on line 8 of SCR # 18, eight numbers are put on the stack, and then taken off by the word SPRITE to create a player called MAN. Of course the eight numbers define the shape of the player - the first one, 18 is the top of its "head" (note the reverse order). Similarly the sequence

```
FF  FF  FF  FF  FF  FF  FF  FF  SPRITE BLOB
```

will create a player-shape called BLOB.

To display MAN on the screen (having first set up the playfield) all you need to do is type 0 MAN. The zero indicates that the shape MAN is to become player zero.

I have modified the Team Atari SCR # 3A so you can use your own players, created using SPRITE, and it now appears as SCR # 19, (by using 3A 19 COPY). The modifications you will need to make are:

1. Remove 0 VARIABLE 0VP from line 1 (it is needed on SCR # 18).

2. Delete the last line of SPLAY (which got the Team Atari shape from SCR # F) and replace it with

   ```
   64   0VP  !    0   MAN
   ```

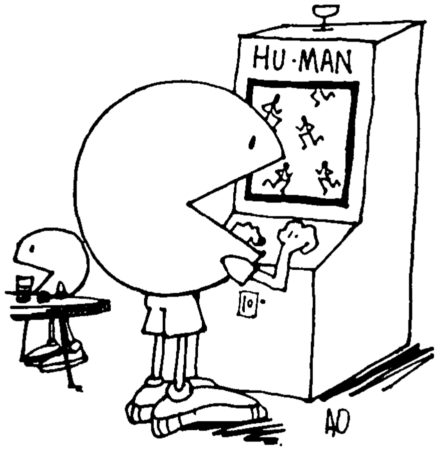
   (this gets the MAN shape).

You may have noticed I changed VPOS! by replacing "AC00" with "Qbase @ 400 +" and "9800" by "Qbase @". This is because the original code would only work properly on a 48 K system - "Team Atari" is a bit flaky in places!

The word SPRITE is interesting because it is a "defining word" - that is a word which creates a family of similar words (in this case "players"). Briefly, the <BUILDS part creates a dictionary entry for the name following SPRITE (e.g. MAN or whatever) and then takes 8 numbers from the stack and compiles them into the dictionary (the "8 0 DO C, LOOP" does this). Now the shape of MAN is permanently associated with its name. The DOES> part is executed whenever MAN (or any other word defined using SPRITE) is executed. DOES> firstly leaves the address of the parameter field of MAN (or whatever) which points to the shape bytes of MAN. Then the code following the DOES> is executed. This calculates the address in the player table where the shape bytes have to go (SWAP 100 * 400 + Qbase @ + 0VP @ +) and the 8 CMOVE transfers 8 bytes from the address left on the stack by DOES> to the playfield address and away we go (after typing RUNIT)!

You can return to FORTH by pressing any key, define a new player shape on the spot (in the text window) using SPRITE and replace the old shape with the new one (0 BLOB, 0 GUN ....).

```
18
( PLAYER MISSILE )
   0 VARIABLE OVP


: SPRITE <BUILDS 8 0 DO C, LOOP
   DOES> SWAP 100 * 400 +
   Qbase @ + OVP @ + 8 CMOVE ;

 42 24 18 18 18 24 5A 18 SPRITE
   MAN




















-->

```



```
19
( PLAYER/MISSILE DEMO-RZ 1/1 )
 64 VARIABLE OHP
 0 VARIABLE OVPOLD
: SPB   HIMEM @ 1+ F800 AND 800
        - DUP Qbase ! 17F +
        HIMEM ! ;
: GETPS OVP ! ROT BLOCK ROT +
        Qbase @ 400 + OVP @ +
        ROT CMOVE ;
: SPLAY STOF 0 0 HPOS! 7 GR.
        SPB Qbase 1+ C@ PMBASE
        C! 2A 0 COLPM! 0 0
        SIZE! 3E D400 C! 3E
        DMACTL C! 3 GRACTL C!
        64 OVP ! 0  MAN ;
: CLRPM Qbase @ 800 ERASE ;
: MOVEH 0 STICK F XOR C AND DUP
         IF 2 / 3 - ENDIF OHP @
         + DUP OHP ! 0 HPOS! ;
: VPOS! OVPOLD @ Qbase @ 400 + +
    DUP Qbase @ 8 CMOVE 8 ERASE
    Qbase @ 400 + + Qbase @ SWAP
    8 CMOVE ;
: MOVEV 0 STICK F XOR 3 AND DUP
         IF 2 * 3 - ENDIF -DUP
         IF OVP @ DUP OVPOLD ! +
         DUP OVP ! VPOS! ENDIF ;
: RUNIT BEGIN MOVEH MOVEV 2FC
         C@ FF = NOT END ;
6 GPRIOR C! ;S


```

* * *

Meeting Dates for 1984

*6th February
5th March
*2nd April
7th May
*4th June
2nd July
*6th August
3rd September
*8th October (due to Labor Day Holiday)
5th November
*3rd December

* indicates release dates for INSIDE INFO subject to the whims and fancies of the new Editor. As there is no meeting in January, please ensure all submissions for February's issue are posted to reach The Editor by 1st January, 1984.

OVERSEAS MAIL ORDERING

by Chris Fitzgerald and Garry Francis

This article will examine the nitty gritty details of buying ATARI software and accessories from overseas - principally the U.S.A. It draws heavily on our personal experiences, but do not quote us on prices or legal aspects as these are constantly changing.

Why Order from Overseas?

There are several reasons why you may want to buy goods from overseas. The most likely one is to save money, but don't count on it! You will have to pay for packing and return postage and possibly import duty and sales tax. These will make your purchase almost as expensive as if you had bought it here in the first place.

A second reason is that the goods you require just aren't available in Australia! Have you seen all the upgrade chips, new disk drives and plethora of software that we constantly read about in the magazines? They can only be bought from overseas!

Deciding What To Buy

Assuming that the item you wish to buy is not available in Australia, you may have to buy it sight unseen. Don't let glossy advertisements and colourful descriptions sway your judgement. Wait until you've read a few unbiased reviews in the computer magazines before making the decision to buy. In the case of software, you can read the reviews from "The Book of ATARI Software 1983" (which is currently on sale at the Pocket and Technical Bookshop and also in our library). Alternatively, you can make your decision based on the software demonstrations at meetings. We have shown a lot of software that's been bought overseas and much of it is unavailable in Australia.

Selecting A Dealer

Once you know what to buy, you'll have to decide who to buy it from. In some circumstances, you won't have any choice. Small add-ons and gadgets and some software are only available from the manufacturer. On the other hand, "big name" products are usually available from a multitude of retailers and mail order houses. Retailers charge normal retail prices and usually don't have very good (if any) mail order facilities. On the other hand, mail order houses have low overheads and hence lower prices - as much as 30% lower! A good mail order house can also fill your order and have it in the post within 24 hours! But be warned that some mail order firms may not think twice about ripping you off! Common tactics include advertising items that aren't yet available, not carrying goods in stock until you actually order them and sending you a different item if the one you ordered was not in stock. Ask around at a meeting to find out which firms are reputable and which are not.

Once you've selected a dealer, send for a price list or catalogue. Then buy only one or two items on your first order. These early experiences will enable you to determine how honest and reliable the firm is. Don't send large amounts of money under any circumstances unless you are confident of the firm concerned.

How Much Do You Send?

Once you have established which firm you wish to deal with, you need to work out how much money to send to cover your goodies. One very important point to remember is that, at present, U.S. dollars cost more than Aussie dollars! The exchange rate varies from day to day, but is currently around $A0.90 to $US1.00. In other words, multiply the cost in U.S. dollars by 1.1 to see what you'll have to pay in Aussie Dollars - and remember it'll always be more! Sorry.

You should also allow enough extra money for the firm to cover the cost of packaging and return postage. Even though their ads may state that packaging and mailing are included, this usually refers only to the country of origin. Surface mail is cheap, but takes 6 to 8 weeks to arrive. You're much better off with airmail, which takes only 1 or 2 weeks, but is far more expensive. We generally allow $US3.00 for airmail postage on a disk or cassette and $US5.00 for a cartridge. Books are generally much heavier and will cost $US8.00 to $US15.00.

Methods of Payment

Money Order:

This is available from the "banking section" of your Post Office. You'll have to fill out a form and sign a declaration to state that the money is not for the payment of goods.

Money Orders can be fairly expensive. Apart from the cost of the amount you want to send, you'll be up for $3.20 commission for the first $40.00 and 8c per dollar (or part thereof) for anything more. The exchange rate at the Post Office is also higher.

A major disadvantage of the Money Order is that you'll get nothing to send with your letter as proof of payment. The Post Office sends your money separately. You'll immediately be overwhelmed by a feeling of vulnerability, as no one on this earth is going to send you your goods until they've seen the colour of your money. You have to place too much trust in the Post Office and the dealer for our liking.

International Draft:

This is the normal method of payment. Go to the Overseas Transactions or Enquiries counter of your bank and tell them that you'd like an International Draft in U.S. currency. At some banks, the first question they'll ask is "Do you have an account with us?". If your answer is no, then you may be out of luck.

You will have to fill in an application form giving details of your name and address, the beneficiary's name and address and the amount, so have all these details beforehand. The teller will make out a draft to your requirements. This is virtually the same as a normal cheque, except that it will be in U.S. currency payable at a U.S. bank.

When you get your International Draft, remember to check that all details on the draft are correct. (The bank once forgot to sign one of our drafts!) Your signature is only required on the application form, not the draft itself.

Get a copy of the application form for your own record (it is the only proof of your transaction) and make sure that the draft serial number has been written on it. Enclose the original draft with your order before mailing.

An International Draft costs $2.00 at the National Australia Bank, whereas Westpac and the Commonwealth charge $3.00 and ANZ charges $4.00. Note that smaller branches may take a lot longer to make out your draft than the larger, city banks, since they may not be familiar with the system.

Credit Card:

In the near future, this may be the most convenient method to pay for your goods from overseas. You will notice in most U.S. mail order ads that you can supply a credit card number rather than pay by cheque. The most popular are Visa and Master Card. Both will be available through certain banks in February to March 1984. Visa is presently available only through the Australian Bank Ltd., but next year it will be available through the ANZ and State Banks. Master Card will be available through Westpac, Commonwealth and National Australia Banks.

It wasn't possible to get many details on either of these cards, but in both cases it will be possible for the overseas dealer to instantly verify your credit number if they care to make a phone call. We were also assured that no company would risk charging more than they were supposed to. It is not known whether there will be an annual fee to own a card, what charges there will be for its use or if you will be required to fill out any customs declarations. It appears that the system is still being worked out by the individual banks, so we can only wait and see.

When you use this method of payment, check to see that the firm doesn't have a surcharge for the use of credit cards.

Sending Your Order

Firstly, make sure you put a return address on all your mail!

Surface Mail:

This costs only 45 cents for the standard letter (Post Office preferred size, weighing up to 20 grams), but will take about 2 months to get to the U.S.A.

Airmail:

It will cost you 75 cents to send a standard letter to the U.S.A. via airmail and will only take about a week to arrive.

Registering Letters:

You can register your letter for an extra $3.30. Registered articles must be signed for by the receiver. This means that you can be pretty sure that the letter will arrive. Note however that you can't certify overseas mail.

Receipt of Acknowledgement:

For yet another 65 cents, you can get a Receipt of Acknowlegement. This is only available for registered articles and is in the form of a card which will be attached to your letter. The receiver must sign the card upon receipt of your letter and the U.S. Post Office sends it back to you. This is handy if you want to be absolutely certain that your letter was received, especially if you sent a lot of money.

When There is No Reply

Boy, have you got problems! You can usually expect goods to arrive around two weeks after mailing (assuming airmail both ways), but wait for four weeks before following up on an overdue order.

There are a couple of things that can go wrong with your order:

- The dealer can do a moonlight flit with your money - always the first thought, but it hasn't happened yet.
- Your letter could have been lost or delayed on the way over.
- The goods could have been lost or delayed on the way back.
- You didn't send enough money to cover airmail postage and the goods have been sent via surface mail.
- There may be a delay in filling your order due to the goods being out of stock.
- The goods may not yet be available, e.g. the RANA disk drives have been advertised for ages in the States, yet they have only just been released - and not without problems. Adventure International's "Sword of Sisyphus" was advertised 12 months before it was available.

At any rate, if your goods do not arrive within a reasonable time, here is what you should do. If you sent a money order, take your receipt to the Post Office and for 95 cents you can get either an Advice of Payment, a Revalidation or have the payment stopped.

If you sent an International Draft, take your receipt along to the bank of issue and ask them to stop payment. You will have to sign an indemnity form, which virtually states that you're not trying to rip anyone off. If the cheque was cashed, the bank will notify you of when and where it was cashed after about a week. If the cheque has not yet been cashed, you may have to wait one or two months before the bank will hand back your money (since your cheque may never be presented).

When the bank pays you, they will re-purchase your original cheque at that day's (buying) exchange rate. If your cheque is returned to you, it is merely a matter of cashing your cheque for Australian money. The National Australia Bank makes no charges for any of these services, but other banks may be different.

Apart from stopping payment of the cheque, you should also write to the dealer again to check on what has happened. You could register the letter and get an Acknowledgement of Delivery to ensure that it arrives.

Import Duty

When your mail arrives from overseas, it goes to the Redfern Mail Exchange for examination by Customs. Any computer related articles will be put aside and examined separately by the Senior Inspector. He is a computer owner himself, so you can't try pleading ignorance or pulling the wool over his eyes. He is well aware of the type and value of the stuff you are importing.

Import duty is payable on most things you would import for your ATARI. To estimate how much duty you may have to pay, you must convert the U.S. value of your goods to Aussie dollars and multiply the resultant figure by the rate of duty payable. There are thousands of different categories for which different rates apply, but the following section should give you a guide. When you've worked this out, add it to the original amount and apply 20% sales tax. The amount that you'll have to pay to Customs is the amount of Import Duty PLUS the amount of Sales Tax.

Duty Payable

The purpose of import duty is to bring the price of imported goods up to the same price that you would expect to pay if they had been made in Australia. The idea behind this is to protect our local industries. It could be argued that as there is no local industry producing ATARI products, that we are being ripped off. True. But other computer related products are made locally, so it is these that are being protected. We are also told that even if there was no local computer industry, the potential exists, therefore the potential industry should be protected. Our industrial laws work in strange ways!

Import duty on computers is 5%. Other hardware such as disk drives and printers are generally 2%. Spare parts are generally the same rate as the item that they are meant to fit in. For example, a new component board for a computer would be 5%, but a similar board for a disk drive would be 2%. Notable exceptions are single integrated circuits which are 2% and bare printed circuit boards with no components which are 35%. Software is also a whopping 35%, but there is no import duty (or sales tax) on books.

You cannot wangle your way out of the 35% duty on software. If your software consists of a disk worth $5 and a printed manual worth $95, you will still have to pay duty on the full amount of $100. The reasoning behind this is that the software constitutes both the program (on disk or cassette) and the documentation, as neither can be used without the other. It is however quite legal to have your media and documentation sent (and invoiced) separately, in which case you will only have to pay duty on the value of the disk or cassette, but not the printed matter.

If you only import small packages of say one or two programs at a time, then it is unlikely that you will get hit by Customs.

A Final Note

Just about the only thing that we haven't mentioned is that if you send for electrical goods (i.e. disk drives, printers, etc.), the U.S. has a different power supply to us, so be sure that you can alter the power requirements to suit the Australian power supply. With ATARI products, the U.S. adapter units are no good in Australia, but they are readily replaced by adapters available locally.

Finally, don't buy an ATARI computer in the U.S.A. as there are two different versions available. The Yanks use NTSC, whereas we use PAL. You cannot buy a PAL system in the States. If you buy an NTSC version, you'll be up for all sorts of extra costs when you get it home - including an NTSC monitor.

HAPPY COMPUTING!

BASICS OF ASSEMBLY LANGUAGE

by Craig Pizzell and Ken Shiu

This series of articles will attempt to help people starting out or interested in Assembly Language. The explanations will be made as simple and straight forward as possible.

Opcodes:

Opcodes are instruction codes used by the programmer which, when assembled, are translated into machine code. Opcodes are identified by a three letter mnemonic. The most commonly used instruction is the LDA instruction, which means LoaD the Accumulator. This instruction places a one byte number (i.e. 0 to 255) in a 6502 register called the accumulator. The accumulator is used to temporarily hold and manipulate data. (Just as a point of interest, there are other registers known as X, Y and Status. We will discuss them later.)

We will compare an LDA instruction to BASIC using LDA #10 as an example. In BASIC, it would be similar to A=10, except that A is a variable. (You don't have to use A. Any variable will do.) The "#" symbol indicates an immediate operand, which simply means load the accumulator with the decimal number 10. If the "#" was omitted, it would mean load the accumulator with the contents of decimal address 10. In BASIC, it would be similar to A=PEEK(10).

STA is the reverse of the LDA instruction. This will STore the contents of the Accumulator into the address specified after the instruction. The STA instruction has the same effect as POKE in BASIC. Here is an example:

```
Assembly Language: LDA #0  ;Load accumulator with 0
                   STA 710 ;Store contents of accumulator into location 710

BASIC:             A=0:REM Load variable A with 0
                   POKE 710,A:REM Store contents of variable A into location 710
```

The semicolon indicates a comment. Comments are used by the programmer to make the program easier to read and therefore easier to debug.

The example below is a little more complex. It will change your screen colours to black on white and make the cursor invisible as soon as something is printed.

```
Assembly Language: 10  *=$600  ;Begin program in page 6
                   20  LDA #0  ;Load accumulator with 0
                   30  STA 709 ;Store at address 709
                   40  LDA #14 ;Load accumulator with 14
                   50  STA 710 ;Store at address 710
                   60  LDA #1  ;Load accumulator with 1
                   70  STA 752 ;Store at address 752
                   80  .END

BASIC:             10 POKE 709,0
                   20 POKE 710,14
                   30 POKE 752,1
                   40 END
```

The above Assembly Language listing is called the source code. This is the human readable form of an Assembly Language program. Note that there are two blanks between the line number and the opcode. To run the sample program with the Assembler/Editor cartridge, type in the source listing, then type "ASM". The ASM command will translate your source program into object code. This is the machine readable form of the code. Next type "BUG", then type "G600" to execute your program. The 600 is your starting address from the sample program (decimal 1536). In DEBUG mode, all numbers are expected to be hexadecimal.

New symbols have been introduced in the sample program. These provide instructions for the assembler, but are not translated into machine language. *= tells the assembler where to begin storing the object code. On other assemblers, this could be ORG for ORiGin, e.g. 10 ORG $600. The $ sign before the address means the number following is a hexadecimal number. Therefore, line 10 in the program tells the assembler that the starting address of the program is to be hexadecimal address 600. The .END tells the assembler that it has reached the end of the source code. The line numbers are also for the benefit of the assembler. Many assemblers do not use line numbers at all.

Look over what we have covered and try using the information in your own simple programs. We have only talked about a few opcodes in this article. In the next article, we will study some more.

* * *

BOOK REVIEW

Introductin To FORTH

Reviewed by Gregg Ramsey

This book is written by Ken Knecht, a freelance author, broadcast engineer and TRS-80 programmer. It is designed specifically for a version of FORTH known as MMSFORTH for the TRS-80, but Knecht claims that most of the examples can be adapted to run on different versions.

From the ATARI programmer's point of view, I think it is of more value to the experienced FORTH user because it has quite a few words that are not included in the standard fig-FORTH. The examples that come to mind are the string handling functions. These include most of the usual Microsoft functions like LEFT$, RIGHT$, MID$ and INSTR().

For someone starting off with FORTH, I think there are better books, like "Starting FORTH" by Leo Brodie. Brodie's book doesn't concentrate on one specific implementation of the language, but covers the universally accepted fig-FORTH. It would suit the Atarian who has just obtained a copy of the public domain fig-FORTH from the Software Exchange and doesn't need all the extra work of weeding out parts that simply don't apply.

The format of "Introduction to FORTH" is very good. The reader is not required to have any previous knowledge of FORTH, reverse polish notation or stacks. Instead, it is assumed that the programmer knows a little about Microsoft BASIC. Just about every FORTH program has an equivalent BASIC program next to it. This helps if the FORTH version is hard to grasp. The screen editor commands, I/O commands and others are quite different to those found in fig-FORTH.

In summary, "Introduction To FORTH" is more formal than Brodie's book (which is liberally sprinkled with humour) and is set out in a clean, logical order. I must emphasize that it has been written for a version of FORTH different to those available for the ATARI, so I cannot recommend it for a beginner. For the person with some experience in FORTH, this book may offer new words and ideas that can be very useful. It also explains things in a different way to other books, so that a concept that was hard to visualise in another book, might just click when you read Knecht's explanation of it.

* * *

FOR SALE

Cartridges: Star Raiders, Chess, Pacman, Assembler Editor
Disks: MAC-65 Macro Assembler, BASIC A+, APX fig-FORTH, Mouskattack, Apple Panic, Threshold
Cassettes: Giant Slalom, Shaft Raider, Space Chase, Lunar Lander
All approximately half price or less in original packages with instructions.
ATARI 822 thermal printer with manual, power pack, connecting cable (does not require 850 Interface Module) and screen dump software. Excellent condition. $200.
Contact Tony Dinallo (02) 662-7777 Ext. 63 (Bus. Hours)

COMPUTER WAVE

ATARI® — THE NEW GENERATION

AVAILABLE NOW!

$399.00 RR

ATARI 600XL

5% DISCOUNT
TO A.C.E. MEMBERS

COMING SOON!

DIG DUG™
The cheerful miner only wants to unearth his beloved fruits and veggies – but first he has to pop the dragons and monsters that guard them.

DONKEY KONG™
Donkey Kong steals Mario's girl, takes her up to his hideout, and dares the intrepid little hero to fight his way up to save her. An instant classic!

AVAILABLE NOW!

ATARI® PAINT

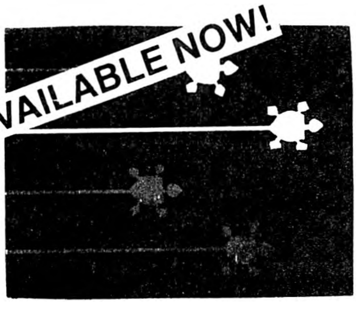
Using this new exciting program called "PAINT" you become the artist – your brush is the joystick, and your canvas is your television set. Create paintings of your own, using the tremendous colour capacities of your ATARI® COMPUTER.

AVAILABLE NOW!

ATARI® LOGO

Plug in an ATARI® Logo Programming Language cartridge, and enter a stimulating environment where mathematical and logical thinking become second nature. ATARI® Logo is so simple that preschoolers can explore profound ideas in words they can understand. Learners of all ages gain insights into geometry, maths or physics using ATARI® Logo's Turtle Graphics.

AVAILABLE NOW!



COMPUTER WAVE
PTY LTD

NOW AT
325 GEORGE STREET
SYDNEY
(between Wynyard Station
and Martin Place)
Tel. 29 1631

STILL AT
LOWER GROUND FLOOR
GRACE BROS STORE
436 GEORGE STREET
SYDNEY
Tel. 231 5791

We believe that our prices are
the best around.

WYNYARD STORE OPEN 7 DAYS A WEEK